IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

SAS INSTITUTE INC., Plaintiff, v. WORLD PROGRAMMING LIMITED; LUMINEX SOFTWARE, INC.; YUM! BRANDS, INC.; PIZZA HUT, INC.; and SHAW INDUSTRIES GROUP, INC., Defendants.	Civil Action No. 2:18-CV-00295-JRG **Jury Trial Demanded**

AMENDED DECLARATION OF BILL ROSENBLATT REGARDING CLAIM CONSTRUCTION OF UNITED STATES PATENT NOS. 7,170,519, 7,447,686, 8,498,996, AND 6,920,458

I, William Rosenblatt, declare as follows:

I. INTRODUCTION

1. My name is Bill Rosenblatt. I am over the age of twenty-one, competent to make this declaration, and have personal knowledge of the matters stated herein.

2. I have been retained on behalf of Plaintiff SAS Institute Inc. ("SAS") to provide expert testimony about, opine on, and analyze United States Patent Nos. 7,170,519 ("the '519 Patent"), 7,447,686 ("the '686 Patent"), 8,498,996 ("the '996 Patent"), and 6,920,458 (the '458 Patent") (collectively, "the SAS Patents"). I have been informed, and it is my understanding, that SAS has accused the following defendants of infringing certain claims of the SAS Patents: Defendants World Programming Limited ("WPL"); Yum! Brands, Inc. ("Yum!"); and Pizza Hut, Inc. ("Pizza Hut") (collectively, "Defendants").

3. My understanding is that the parties dispute the meaning and scope of certain claims and claim terms in the SAS Patents. I further understand that, after I executed my original Declaration on November 12, 2019, Defendants modified their claim construction contentions, including the removal of several disputed claim terms. I submit this Amended Declaration to withdraw my testimony on terms that are no longer in dispute, because that testimony is no longer relevant. No other testimony has changed, and I have endeavored to keep the paragraph numbers unchanged as well. This declaration sets forth my expert opinions about claim construction disputes for those claim terms of the SAS Patents that the Defendants allege are indefinite or means-plus-function terms under § 112(6). I also provide certain analysis and opinions herein about the general subject matter, teachings, and technology of the SAS Patents for purposes of claim construction.

4. My understanding is that I may be called to testify about this declaration, whether at deposition or at a claim construction hearing held by the Court. During the course of any such

testimony, I may refer to or cite information or documents that are similar to those discussed and identified in this declaration. To explain my analysis and opinions while testifying, I also may use PowerPoints, pictures, animations, demonstrations, or other aids in order to explain my testimony, analysis, and opinions.

5. In this case, SAS is compensating me for my time spent on this engagement, as well as reimbursing me for my reasonable expenses. My standard consulting rate is $600 per hour, which is the rate I have charged to prepare this declaration. I do not have any personal stake or interest in the outcome of these cases, and my compensation does not depend on the testimony that I am providing. I am providing all of my analysis and opinions as an independent expert, and regardless of the outcome of these cases. Each opinion I state herein is my own, which I have reached after analyzing the SAS Patents and other materials referenced below.

II. BACKGROUND AND QUALIFICATIONS

6. I have attached as **Exhibit A** to this declaration a copy of my curriculum vitae, which provides a description of my professional experience, including my academic and employment history, conference participation, teaching experience, and more. Also included with **Exhibit A** is a list of the publications I have authored in the past 10 years. In my CV and the discussion below, I set forth and explain my qualifications to act as an expert in the subject matter and technology of the SAS Patents.

7. In formulating my opinions, I have relied upon my training, knowledge, and experience in the relevant art. I received a bachelor's degree *cum laude* in Electrical Engineering and Computer Science from Princeton University in 1983. I was then employed as a software engineer in data communications, secure telephony, and system administration at Motorola and Intermetrics, Inc. (now part of Level 3 Communications).

8. I received a Master of Science degree in Computer and Information Science from the University of Massachusetts at Amherst in 1990, where I conducted graduate-level research in software engineering, programming languages, and databases, and I taught a course in Software Engineering for senior and first-year graduate students.

9. After graduate school, I was employed as Senior Consultant at The Rustin Group, a software development and consulting firm in New York. In that capacity, I developed a database application for an international bartering company. Subsequently, as Manager of Information Resources at Moody's Investors Service, the bond rating agency, I helped develop a digital publishing system that used database queries to extract bond ratings data from Moody's ratings database and embed it in digital publications that would be delivered electronically to Moody's customers.

10. In 2000, I founded GiantSteps Media Technology Strategies, the consulting firm that I run to this day. I have had regular experience since then in writing data querying and other software in a number of languages. In the early 2000s, I created a concept called Enterprise Content Integration, a way of integrating multiple databases containing digital media content throughout a large enterprise without having to migrate them all to the same type of database; I consulted on Enterprise Content Integration to the Associated Press among others.[1]

11. In addition, I have had experience with computer graphics dating back to my time as an undergraduate in the early 1980s, when I took a course and did my junior year independent project in the subject. I have also had education in statistics as part of my undergraduate and graduate coursework in computer science and mathematics.

[1] *See for example*, Rosenblatt, B.: Enterprise content integration: a progress report. Seybold Report: Analyzing Publish Technologies 3, 9–14 (2003).

12. During the course of my professional employment, I have used numerical analysis and graphics software such as Microsoft Excel extensively, and graphical display of numerical data has featured heavily in articles I have written on the media industry for *Forbes*.[2] Currently I teach the course Data Analysis in the Music Industry at New York University, where the subject matter includes data visualization (the graphical representation of data and information) as well as data querying.

13. I have served as an expert witness in several litigations concerning computer and software technology. I have given testimony in the following matters over the past four years, on behalf of parties identified in **boldface**:

- *Marcus Gray v. **Katy Perry**,* C.D. Calif., 2019 (trial);
- *Energy Intelligence Group v. **CHS McPherson Refinery***, D. Kan., 2017 (deposition).

III. MATERIALS CONSIDERED

14. To prepare this declaration, I have considered the SAS Patents, their prosecution histories, and the parties' proposed claim constructions and supporting evidence. I have also considered the materials that are discussed in the body of this declaration. I have also considered certain legal standards, discussed later in this declaration, that I understand apply to patent claim construction.

15. I have prepared this declaration based upon the information and materials that are currently available. In the event that additional information becomes available or is provided to me, then I may change, supplement, or amend my analysis and opinions based on any newly available information. For example, I have not yet received Defendants' claim construction

[2] *See for example*, https://www.forbes.com/sites/billrosenblatt/2019/02/06/spotify-spends-big-on-podcasts-but-ad-revenue-has-a-long-way-to-go/; https://www.forbes.com/sites/billrosenblatt/2018/05/07/the-short-unhappy-life-of-music-downloads/; https://www.forbes.com/sites/billrosenblatt/2018/04/08/in-the-new-era-for-music-streaming-rules-but-human-factors-endure/.

briefing or any expert testimony from Defendants. If Defendants provide claim construction briefing or expert testimony that is related to my opinions stated herein, then I may review this new information and revise my analysis and opinions accordingly.

IV. LEGAL STANDARDS

16. To prepare these claim construction opinions, I have applied my understanding of certain legal standards of patent law related to determining and interpreting claim scope. To be clear, I am not a lawyer or an expert in patent law. I have been informed that it is ultimately the Court that will decide and apply the legal standards related to claim construction. Following is my understanding of these legal standards that the Court will apply in its analysis.

A. General Principles

17. The scope and meaning of a patent claim is determined based on what the claim would mean to a person of ordinary skill in the art ("POSITA"). This determination is made as of the filing date of the patent and based on intrinsic and extrinsic evidence. In my discussion below, I address what I believe to be the level of skill required for a POSITA for the SAS Patents.

18. A patent's claim language and specification are intrinsic evidence. One should analyze the intrinsic evidence from the perspective of a POSITA to understand a claim term. A patentee may define or ascribe a particular meaning to a claim term in the specification. However, a POSITA should not read limitations from the specification into a claim if the limitation is not already found in the claim language.

19. The prosecution history of a patent is also intrinsic evidence. This is the back and forth communications between the patent applicant and the U.S. Patent and Trademark Office (PTO), which creates a record of the patent's prosecution. The prosecution history may indicate how the claims should be interpreted, how the inventor (or the PTO) understood the patent, and

differences between the claims and the prior art. My understanding is that amendments to the claim language or arguments made by the applicant during prosecution can be indicative of or limit the claim scope or meaning in the patent that issues from that application, as well as in related patents.

20. To determine the meaning of a claim term and claim scope, one should read and analyze a claim term in the context of the entire patent in which the term appears, including its specification and prosecution history, and not just in the context of a particular claim. Also, one should consider the education level, knowledge, and experience of a POSITA when determining the meaning of a claim term.

21. Extrinsic evidence can also be considered when determining the meaning of a patent claim, but it is not as important as, and cannot supersede the meaning according to, the intrinsic evidence. Extrinsic evidence can include technical dictionaries and expert testimony.

B. Means-Plus-Function Limitations

22. Sometimes, a patentee may express a claim term as "means" for performing a certain "function," or use similar language. I understand that a claim term in this format is called a "means-plus-function" claim term.[3] When expressed in this format, my understanding is that the claim term does not identify the actual structure of the claim limitation. Instead, the claim term only describes the function performed by the limitation. One must then look to the specification to identify the corresponding structure identified by the patentee for performing the stated function.

[3] Because all of the patents-in-suit were filed prior to the America Invents Act (AIA), and its renumbering of 35 U.S.C. § 112, I use the old statutory notation, "§ 112(6)" or "§ 112 ¶ 6," instead of the post-AIA notation, "§ 112(f)."

23. My understanding is that parties may sometimes dispute whether a claim term is a means-plus-function term. My understanding is that the use of the word "means" provides a rebuttable presumption that the term is a means-plus-function term. However, if the word "means" is not used, my understanding is that there is a rebuttable presumption that a term is not a means-plus-function term.

24. The presumption stands or falls according to whether one of ordinary skill in the art would understand the claim with the functional language, in the context of the entire specification, to denote sufficiently definite structure or acts for performing the function. As a result, my understanding is that when a claim term lacks the word "means," the presumption that the term is not a means-plus-function term can be overcome if the challenger demonstrates that the claim term fails to recite sufficiently definite structure or else recites function without reciting sufficient structure for performing that function. I also understand that one way to demonstrate that a claim limitation fails to recite sufficiently definite structure is to show that, although not employing the word "means," the claim uses similar "nonce words" that can operate as a substitute for "means" because they typically do not connote sufficiently definite structure. In the software context, my understanding is that terms such as "program," "software application," and "user interface code" are not considered to be nonce terms, and that the mere fact that the disputed limitations incorporate functional language does not automatically convert the words into means for performing such functions. Further, my understanding is that if the term itself (or the claim in which the term appears) recites or would be understood to connote adequate structure for performing the claimed function, then the claim term is not a means-plus-function term.

25. If a term is a means-plus-function term, my understanding is there are two steps for determining the meaning and scope of the term. First, one must identify the claimed function set forth in the claim for the term. Second, one must identify the structure disclosed in the written description that corresponds to the claimed function. When there are multiple claimed functions, one must identify the corresponding structure that performs each of the claimed functions. Structure disclosed in the written description is 'corresponding' structure only if the written description or prosecution history clearly links or associates that structure to the function recited in the claim.

26. I understand the corresponding structure must include all structure that actually performs the recited function. However, § 112 does not permit incorporation of structure from the written description beyond that necessary to perform the claimed function.

27. For means-plus-function claims in which the disclosed structure is a computer, or microprocessor, programmed to carry out an algorithm, I understand that typically the disclosed structure is not the general purpose computer, but rather the special purpose computer programmed to perform the disclosed algorithm. The specification can express the algorithm in any understandable terms including as a mathematical formula, in prose, or as a flow chart, or in any other manner that provides sufficient structure.

28. However, absent a possible narrower construction of the terms 'processing,' 'receiving,' and 'storing,' a function with any of those terms can be achieved by any general purpose computer without special programming. In other words, a general-purpose computer is sufficient structure if the function of a term requires no more than merely processing, receiving, or storing, and no algorithm needs to be disclosed.

C. Definiteness

29. A claim must inform a POSITA about the scope of the claims with reasonable certainty. If the scope and meaning of a claim or one of its terms is not reasonably certain, then my understanding is that the claim is indefinite.

30. I understand that a determination of claim indefiniteness is a legal question for the Court to decide. I understand that when a term of degree (e.g., "unobtrusive" or requiring a quantity that could be measured in different ways) is used in a claim, the court must determine whether the patent provides some standard for measuring that degree. Likewise, when a subjective term (e.g., "does not distract a user", "critical to the user") is used in a claim, the court must determine whether the patent's specification supplies some standard for measuring the scope of the term.

31. Claim language employing terms of degree has been found definite where it provided enough certainty to one of skill in the art when read in the context of the invention. It is my understanding that absolute or mathematical precision is not required. That is, the definiteness requirement mandates clarity, while recognizing that absolute precision is unattainable. I further understand that the claims, when read in light of the specification and the prosecution history, must provide objective boundaries for those of skill in the art. A term of degree fails to provide sufficient notice of its scope if it depends on the unpredictable vagaries of any one person's opinion.

32. For means-plus-function claim terms, when no structure in the specification is linked to the function in a means-plus-function claim element, I understand that such a claim is indefinite. I further understand that a means-plus-function clause is indefinite if a person of ordinary skill in the art would be unable to recognize the structure in the specification and associate it with the corresponding function in the claim.

33. When the specification discloses no structure, a POSITA's knowledge is irrelevant, and the claim is indefinite. When the specification discloses some structure, a POSITA's knowledge is relevant to determining whether the disclosure is a sufficiently precise description of the corresponding structure. If the specification does not have sufficient structure for performing the claimed function(s), as would be apparent to a POSITA, then my understanding is that the term is indefinite.

34. My understanding is that the Defendants bear the burden of proving that a claim term is indefinite, and that indefiniteness must be proven by clear and convincing evidence. My understanding is that this standard of proof is an intermediate standard which lies between "beyond a reasonable doubt" and a "preponderance of the evidence." My understanding of the phrase "clear and convincing" is that it describes evidence that produces in one's mind a firm belief or conviction as to the matter at issue.

V. TECHNOLOGY BACKGROUND AND THE SAS PATENTS

A. The '686 and '996 Patents

35. The inventions described and claimed in the '686 and '996 Patents relate to programming languages for relational database systems. Relational databases—based on tables of data—were invented in the 1970s and became the dominant type of database system for large-scale applications in the 1980s. They are still the most widely-used type of database technology today (other types include hierarchical, network, and object-oriented).

36. The standard programming language for manipulating and querying relational databases is Structured Query Language (SQL). SQL is a prominent example of so-called fourth-generation languages (4GLs), which manipulate data at higher levels of abstraction than so-called third-generation languages like C, PASCAL, BASIC, and Java.

37. Since the 1980s, there have been several popular database systems based on the relational model, such as Oracle, Sybase, DB2, and SQL Server. Each of these has its own dialect of SQL with slightly different syntaxes (grammar) and semantics (meanings of language constructs), or proprietary extensions to the standard SQL[4] language.

38. Throughout the 1990s, more and more large organizations adopted multiple relational or relational-like data systems and found the need to interoperate data among multiple types of database systems using different dialects of SQL. For example, if Company A merged with Company B, and the two companies used different types of databases, it would be desirable to figure out how to access Company B's data from Company A's systems and vice versa. In addition, as personal computers became ubiquitous in the 1990s, a new breed of personal relational-like database systems such as dBASE, Clipper, FileMaker, and FoxPro—all of which stored data in tables, as relational databases do--achieved footholds within small workgroups but did not easily integrate with one another across larger enterprises.

39. By the early 2000s, it became clear that it was not practical to replace all the different types of databases within an organization with a single database.[5] Techniques were needed to enable data from one database to be included in queries from another database, so that query results could include data from both databases.

40. An illustrative example is the "join" statement in the SQL language. "Join" causes two tables to be combined using data fields that are common to both. Assume for example that one table in a database at a mobile telephone company contains information about people and their addresses and mobile phone numbers, and another table contains information

[4] Persons of skill in the art, as defined below, have been known to pronounce "SQL" as either "Sequel" or "Ess-cue-ell," the former requiring "a" and the latter requiring "an." I use the latter.

[5] See *supra* note 1.

about phone numbers, features of the calling plan at that phone number, and usage data. It is possible to use the SQL join statement to combine the two tables on the common phone number fields and find out the names and addresses of people who have certain mobile calling plan features—perhaps in order to send those people a brochure in the mail. For example, the company might want to offer an international calling plan to customers who have high numbers of international calls but do not have international calling plans. This is a simple example; there are several different types of joins, some of which are quite complex, and they tend to be implemented with different syntaxes and/or semantics in different types of databases. It was desirable in the early 2000s to be able to join tables on multiple different databases. Hence there is a need to be able to process joins among tables in different databases in a uniform, coherent way.

41. The ’686 and ’996 Patents are both directed to methods and systems that enable interoperation of heterogeneous databases at the query language level. The invention claimed in the ’686 Patent involves a technique for translating statements in different query languages into a common syntax tree representation, to enable translation of queries written for one type of database into queries for another type of database. Queries submitted within one database environment can be translated and then sent to and processed by another type of database. For each type of database, there are techniques for translating between the syntax tree representation and statements in that database’s language. (*See for example* ’686 Patent, 2:18–25).

42. The invention claimed in the ’996 Patent is directed to a similar set of problems to those addressed by the ’686 patent, but it discloses a different technique. The ’996 Patent describes systems and methods for algorithmically translating a query written in one language directly into the (different) language of a target database, using a “control string” selected from a

table that specifies the appropriate syntax of a language construct for the target database. (*See for example* '996 Patent, 5:16–25).

B. The '519 Patent

43. The '519 Patent is also directed to interoperability of data across different types of computer systems. But instead of 4GL query language statements, the '519 Patent is directed to computer graphics—specifically graphics used to display numerical and statistical charts from data. Various types of software, ranging from spreadsheets such as Microsoft Excel and Google Sheets to large-scale statistics and business analytics software packages such as SAS, Business Objects, and Cognos are widely used to display charts and graphs of data.

44. The kinds of charts that these packages can generate include common types such as bar charts, pie charts, maps, and x-y graphs. Just as organizations came to adopt multiple types of database systems in the 1980s and 1990s, organizations also came to adopt multiple types of systems for generating data graphics. There was a need for techniques for describing data charts in consistent ways that enabled the descriptions of the visual formatting of charts to be decoupled from specific database systems and ported from one system to another with the assurance that charts will be displayed consistently. For example, a bar chart has bars, which have widths and colors; it also has axes with line thicknesses, tick marks, text with fonts, and so on. There was a need for a simple way to instruct different data graphics systems to display bar charts with the same visual characteristics, so that a chart generated by one system would look the same as a chart generated by a different system.

45. The '519 Patent is directed to systems and apparatuses for generating data graphics with similar output across multiple systems by defining "graph style data" such as line styles, bar thicknesses and colors, pie chart colors, etc. Graph styles can describe elements such as points, lines, bars, pie slices, grid lines, and axes; or they can relate to roles in statistics such

as category and response variables (see ¶76 below). Graph style data items contain descriptions of how graph styles should be displayed in terms that include line color, text font, text color, text border, line width, and so on. Graph style data can be expressed in languages such as XML (eXtensible Markup Language), a widely used standard for expressing structured data. The '519 Patent describes systems and apparatuses for taking graph style data and generating input to graphics generators that causes them to display data accordingly. (*See for example* '519 Patent, 1:40–45.)

C. The '458 Patent

46. The '458 Patent is directed to the field of *data mining*. Data mining is a set of tools and techniques for analysis of large data sets, to uncover patterns and trends that cannot be detected using simpler analytic techniques. Data mining tools such as SPSS Modeler from IBM, Oracle Data Mining, and SAS Enterprise Miner from SAS Institute enable business analysts and data scientists to sift through large data warehouses and detect patterns that can be used to predict future outcomes. A related field is *machine learning,* which is the use of patterns and statistics to create sets of rules about how to categorize data and make decisions automatically. Data mining and machine learning are both subfields of *predictive analytics*, a field that ties together data mining, machine learning, and statistical analysis to identify the likelihood of future outcomes based on historical data.[6] Although the term "predictive analytics" was not in wide use until the 2000s,[7] predictive data analysis techniques using statistics and machine learning algorithms were well known before then.

[6] https://www.sas.com/en_us/insights/analytics/predictive-analytics.html.

[7] E.g., the Wikipedia page for "predictive analytics" was created in 2006.

47. Analysts who perform data mining tasks develop *specifications* that indicate the data to analyze and the algorithms used to detect patterns. When a data mining tool runs with a specification, it generates a set of information that is referred to as a "model." According to the '458 Patent, the attributes of a data model include the location of the input data, the scoring code, the fit statistics, and so on. '458 Patent, 4:2-24. Over time, users in an organization may develop many specifications that can be reused as the data in the data warehouse changes over time. These are sophisticated and complex bodies of code, and thus are worth reusing wherever possible so as to avoid "reinventing the wheel." The scoring code, fit statistics and the like are valuable because they enable future users of the specifications to judge their accuracy and relevance to other data sets. This leads to the need to manage the models centrally, so that analysts can know which models are available and what types of analytic tasks they are good for instead of "reinventing the wheel" every time.

48. The "model repository" disclosed in the '458 Patent is a way of centrally organizing and managing models used in data mining. To do this, it uses a data structure that organizes models into multiple levels, including project, diagram, and model. A diagram describes a set of model specifications; a project includes a set of diagrams. The model repository contains descriptions of features of models that are organized in ways that enable users to search for models that meet their needs for data analysis.

VI. THE LEVEL OF SKILL IN THE ART

49. I understand that claims are interpreted from the perspective of a POSITA at the time of the invention. To determine the characteristics of a POSITA for the SAS Patents at the times of the inventions, I have considered the state-of-the-art of computer systems, databases, and data graphics in the relevant time period for each of the SAS Patents. This includes

considering the problems encountered in the art at that time. I also have considered the education and experience of persons who were working in the field at that time, including my own experience and knowledge at that time and others who I worked with on these issues and problems. Further, I have considered the experience, education, and knowledge of persons in industry and academia at that time who were working in the aforementioned fields.

50. The ’519 Patent was filed on April 15, 2002 and claims priority to Provisional Application No. 60/368,896, which was filed on March 29, 2002. For purposes of this declaration, I consider the time of the invention to be the March 2002 timeframe when that provisional application was filed. However, my opinions would not change even if I were to consider the time of the invention to be the April 15, 2002 filing date of the non-provisional application that later issued as the ’519 Patent. It is my opinion that a POSITA for the ’519 Patent would have a bachelor’s degree in computer science or a similar field, including coursework in statistics, plus two years’ work experience in the field, or education and work experience equivalent thereto, plus at least one year’s experience in the use of data graphics software (potentially including spreadsheet packages). As stated above and in my CV, I have at least these qualifications—both today and at the time of the invention of the ’519 Patent.

51. The ’686 Patent was filed on November 22, 2002. For purposes of this declaration, I consider the time of the invention to be the November 2002 timeframe when the non-provisional application that later issued as the ’686 Patent was filed. It is my opinion that a POSITA for the ’686 Patent would have a bachelor’s degree in computer science or a similar field, including coursework in database systems and knowledge of multiple programming languages, plus two years’ work experience in the field, or education and work experience

equivalent thereto. As stated above and in my CV, I have at least these qualifications—both today and at the time of the invention of the ’686 Patent.

52. The ’996 Patent was filed on November 3, 2008. For purposes of this declaration, I consider the time of the invention to be the November 2008 timeframe when the non-provisional application that later issued as the ’996 Patent was filed. It is my opinion that a POSITA for the ’996 Patent would have a bachelor’s degree in computer science or a similar field, including coursework in database systems and knowledge of multiple programming languages, plus two years’ work experience in the field, or education and work experience equivalent thereto. As stated above and in my CV, I have at least these qualifications—both today and at the time of the invention of the ’996 Patent.

53. The ’458 Patent was filed on September 22, 2000. For purposes of this declaration, I consider the time of the invention to be the September 2000 timeframe when the non-provisional application that later issued as the ’458 Patent was filed. It is my opinion that a POSITA for the ’458 Patent would have a bachelor’s degree in computer science or a similar field, including coursework in database systems, plus two years’ work experience in the field, or education and work experience equivalent thereto. As stated above and in my CV, I have at least these qualifications—both today and at the time of the invention of the ’458 Patent.

VII. CLAIM CONSTRUCTION ANALYSIS OF THE PATENTS-IN-SUIT

54. I have been asked to opine on the meaning of certain disputed claim terms of the SAS Patents. Specifically, I have been asked to opine on whether Defendants have established that certain disputed claim terms are indefinite, and whether these terms are, in fact, indefinite, according to the understanding of a POSITA. For claim terms that Defendants have argued are means-plus-function terms under § 112(6), I have also been asked to opine on whether,

according to the understanding of a POSITA, those terms should be construed as means-plus-function terms. The following is my analysis for these claim terms. In my opinion, each of the following claim terms is definite, and Defendants have not presented evidence or otherwise proven that any of these terms are indefinite by clear and convincing evidence. In my analysis, I first analyze the claim language, specification, and extrinsic evidence for each allegedly-indefinite claim term. I then address Defendants' contention the claim term is indefinite, explaining why the Defendants are incorrect and that the claims inform, with reasonable certainty, a POSITA about the scope of the invention.

55. In my opinion, Defendants have failed to show by clear and convincing evidence that any claim term I discuss below is indefinite. In other words, in light of the intrinsic and extrinsic evidence, the claim term informs a POSITA about the scope of the claims with reasonable certainty, and there is no clear and convincing evidence otherwise.

A. The '519 Patent Claims and Disputed Claim Terms

56. The following are representative asserted claims from the '519 Patent. The claim terms that Defendants assert are indefinite and/or governed by § 112(6) are shown in ***bold italics***. Claim terms that Defendants assert are governed by § 112(6) are also <u>underlined</u>.

> 1. [pre] A computer-implemented method for generating data graphical displays, comprising the steps of:
>
> [1.1] receiving data to be displayed in a ***non-textual format***, said received data being indicative of a plurality of variables;
>
> [1.2] retrieving graph style data items from a data file, said graph style data items containing display characteristics to be used in displaying the data in a ***non-textual format***; and
>
> [1.3] accessing of the graph style data items in order to display ***non-textual formatted output*** based upon the graph style data items;

[1.4] said graph style data items containing graph style metadata that have descriptors specifying what statistical roles different data variables have within the data;

[1.5] wherein the specified statistical roles are used to define display characteristics for the data;

[1.6] wherein the data is displayed in a ***non-textual format*** in accordance with the graph style data items and the graph style metadata.

* * *

34. [pre] A computer-implemented apparatus for generating data graphical displays based upon data, comprising:

[34.1] a ***graph generator module*** that receives data to be displayed in a ***non-textual format***, said received data being indicative of a plurality of variables;

[34.2] graph styles data structure that defines display characteristics to be used in displaying the data in a ***non-textual format***,

[34.3] said graph style data structure containing graph style metadata that defines display characteristics for data through the metadata associating at least two of the variables with statistical roles;

[34.4] said ***graph generator module*** having data access to the graph style data structure,

[34.5] said ***graph generator module*** generating at least one graphical output based upon the received data, said graphical output being generated in accordance with the defined data characteristics of the graph styles data structure.

* * *

37. The apparatus of claim 34 ***wherein the graph style metadata identifies a data variable as having a category role and identifies another data variable as having a response role, said category role and said response role being used by in different output non-textual formats.***

The following is my claim construction analysis for the terms above. As discussed below, none of these terms is indefinite, and a POSITA would not have a firm conviction based on the intrinsic and extrinsic evidence that any of these terms are indefinite.

i. “non-textual format” / “non-textual formatted output”

Term (Claim)	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“non-textual format” / “non-textual formatted output” (claims 1, 2, 5, 6, 7, 8, 9, 12, 14, 15, 16, 17, 33, 34, 37, 38, 39, 40, 41, 42, 43, 44, 45, 50, 58)	Plain and ordinary meaning	Indefinite

57. The claim terms “non-textual format” and “non-textual formatted output” appear throughout many of the asserted claims. The meaning of each of these terms is clear and definite. “Non-textual format[ted output]” refers to a format by which the output is not displayed as text, such as in a graphical format.

58. The meaning of these claim terms is clear and definite from the claim language itself. For example, in claim 34, the term “non-textual format” appears in the context of limitations [34.1]–[34.5]. These limitations describe a “graph generator module that receives data to be displayed in a non-textual format,” “graph styles data that defines display characteristics to be used in display the data in a non-textual format,” and a “graph generator module [that] generat[es] at least one graphical output” (*Id.*, claim 34, limitations [34.1]–[34.5]), which all clearly indicate that “non-textual format” refers to data that is to be displayed graphically or in some form other than a textual format.

59. It is also clear to a POSITA that “non-textual format” does not just refer to a graphical format, because claim 2 of the ’519 Patent recites “The method of claim 1 wherein the

non-textual format includes a graphic format." That is, claim 2 further limits claim 1 by specifying a graphical format, as opposed to another type of non-textual format.

60. Further evidence of the clarity of the meaning of "non-textual format" and "non-textual formatted output" comes from the specification. When describing the background and summary of the invention, the '519 Patent explains that the invention relates to "[g]raphical depictions of computer-generated data" and the "appearance of graphical displays" and associated "graphical styles." ('519 Patent, 1:26–36). The specification then describes the advantages of the invention, including that data that is displayed in a "non-textual format" refers to data displayed graphically or in some other non-textual representation:

> The present invention overcomes the aforementioned difficulties by providing a computer-implemented system and method for generating ***data graphical displays***. The data is indicative of a plurality of variables. The system and method receive data that is to be displayed in a ***non-textual format. Graph style data items*** are retrieved that ***contain the display characteristics for displaying the data in the non-textual format. The data is displayed*** in accordance with the graph style data items.

(*Id.*, 1:37–45, emphasis added.)

61. The detailed description of the '519 Patent also makes it clear that "non-textual" output refers to graphic or other type of output:

> The definition and generation of ***graphic output*** via the ***graph styles data*** may be performed in many different ways. … For example, such unique ***graph styles*** as the following may be predefined: an analysis style (e.g., showing a magnifying glass in the background); an astronomy style; a banker style; a blockprint style; a convention style; a curve style; an education style; an electronics style; a gears style; a magnify style; a money style; an RSVP style; a science style; a sketch style; a statistical style; a torn style; a watercolor style; and others. These styles may allow the following ***display characteristics*** to [be] used with ***non-textual output***: soft shadows around text; transparency of data primitives, legends and graphics area; texture maps on data primitives and graphics area; background images specific to industries; use of specific fonts to highlight labeling and values; color scheme to

> enhance the overall appearance of the graphs; gradient fills for the graphics area; linestyles to highlight data; and others.

(*Id.*, 6:4–30, emphasis added.)

62. The foregoing makes it clear that "non-textual" refers to graphical aspects, or "display characteristics," of visual output that can be represented as "graph styles data": shadows around text, transparency of displayed text (i.e., how much of the background shows through the text), background images, colors, gradient fills, highlighting, and so on. This is nothing more or less than the plain and ordinary meaning of "non-textual" as understood by a POSITA.

63. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that these terms are "[i]ndefinite," but do not provide any explanation for why the terms are allegedly indefinite. This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how it is not clear what non-textual format and non-textual formatted output means, and how the format and/or formatted output can be text." This statement is similarly unclear about Defendants' contention why these terms are indefinite, and they are not indefinite as explained above.

64. For the reasons discussed above, it is my opinion that a POSITA would find that the claim terms "non-textual format" and "non-textual formatted output" reasonably inform the POSITA about the scope of the claim and are not indefinite.

ii. [withdrawn]

[Paragraphs 65-74 withdrawn]

iii. "wherein the graph style metadata identifies a data variable as having a category role and identifies another data variable as having a response role, said category role and said response role being used by in different output non-textual formats"

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"wherein the graph style metadata identifies a data	Plain and ordinary meaning	Indefinite

variable as having a category role and identifies another data variable as having a response role, said category role and said response role being used by in different output non-textual formats” (claim 37)		

75. This claim term appears in dependent claim 37. The meaning of this claim term is clear and definite, including from the claim language itself. Claim 37 states that the “graph style metadata” (i.e., metadata describing the graph style of a variable) performs two identifications. First, this metadata “identifies a data variable as having a category role.” (’519 Patent, claim 37). A POSITA would understand what a “category role” is in the relevant art and that a variable can be a category variable, i.e., can have a category role. Specifically, a POSITA would understand a category variable to be a variable that can take on one of a limited number of values, i.e., a “bucket” such as a spoken language, political party, or demographic affiliation.[8] (*See for example* ’519 Patent, 2:63–64, 4:62–64, 5:13–19, 5:45–49). Second, this metadata “identifies another data variable as having a response role.” (*Id.*, claim 37.) A POSITA would similarly understand what a “response role” is and that a variable can have a response role. Specifically, a POSITA would understand a response variable to be a variable that is dependent on an input (or independent) variable, such as an outcome of an experiment or mathematical function.[9] (*See for example id.*, 3:2–4, 3:30–35, 4:64–66, 5:22–24). Defendants do not assert that either of these portions of claim 37 are indefinite, nor do they assert that “category” or “response” alone are indefinite.

[8] Many statistics texts use the term “classification” instead of “category” variable; a POSITA would understand that these terms are equivalent.

[9] “Dependent variable” is an alternative term for “response variable”; *see* ¶71 above.

76. Instead, Defendants appear to assert that the last portion of claim 37 is indefinite due to an obvious typographical error. This last portion states, "said category role and said response role being used ***by in*** different output non-textual formats." (*Id.*, claim 37, emphasis added). This obvious typographical error does not render claim 37 indefinite, and even as written claim 37 reasonably informs a POSITA of its claim scope in light of the specification. This language appeared in the original application to the PTO, and the Examiner never raised an issue with it despite its appearance in several rounds of amended claims (*see for example* '519 File History at 4/15/02 Application p.27; 7/13/04 Responsive Amendment p.9; 1/19/05 Responsive Amendment p.9; 5/25/05 Responsive Amendment p.9; 1/13/06 Responsive Amendment p.9; 6/30/06 Responsive Amendment p.9). Regardless of whether the drafter meant to say that the category role and response role are used "by" or "in" different output non-textual formats, the meaning of this phrase is the same: data having a category role and data having a response role—as identified by the graph style metadata—is used when displaying that data in bar charts, pie charts, graphs, scatter plots, and so on (i.e., "by" and/or "in" the "different output non-textual formats"). Put another way, the claim language is clear that the category role and the response role are used to generate outputs in different non-textual formats. In particular, this is explained by the specification, which discloses that data for category and response variables may be displayed differently, according to different output formats. (*Compare id.*, *e.g.*, Fig. 5 and 5:27–55 (displaying year "category" variable data and sales "response" variable data in one manner for a bar chart output format) *with* Fig. 6 and 5:56–63 (displaying same year "category" variable data and same sales "response" variable data in a different manner for a pie chart output format). *See also id.*, 3:12–14 ("An aspect of a graph may be formatted based upon a combination of graph styles data associated with a variable and the variable's role.")). In light of

these portions of the specification, a POSITA would understand the meaning of claim 37, which is not indefinite in spite of the minor and obvious typographical error "by in" identified by Defendants.

77. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that this term is "[i]ndefinite". This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding this term regarding how it is unclear as to whether "said category role and said response role" are used "by"/"in" "different output non-textual formats." Based upon this statement, Defendants appear to assert this term is indefinite due to the "by" / "in" typographical error. As discussed above, however, that obvious and minor typographical error does not render this term indefinite.

78. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "wherein the graph style metadata identifies a data variable as having a category role and identifies another data variable as having a response role, said category role and said response role being used by in different output non-textual formats" is not indefinite.

iv."the graph generator module"

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"the graph generator module" (claims 34, 52, 53, 58)	Plain and ordinary meaning 35 U.S.C. § 112(6) does not apply	Indefinite 35 U.S.C. § 112(6) applies. Function(s): receives data to be displayed in a non-textual format; having data access to the graph style data structure; generating at least one graphical output based upon the received data Structure: none

79. The claim term "the graph generator module" appears in independent claim 34 and its dependent claims 52 and 53, as well as independent claim 58. The meaning of this term is clear and definite. "The graph generator module" refers to the hardware and/or software that generates a graph. Defendants assert that a POSITA would understand that this term should be construed as a means-plus-function term (a "112(6) term") lacking corresponding structure and is therefore indefinite. I disagree that a POSITA would understand this to be a 112(6) term. A POSITA would understand a "graph generator module" to be a well-known software component that is used to generate graphs, akin to a device driver for a plotter (a hardware component used to create hard copies of data graphics). The operational details of the graph generator module are not germane to a POSITA's ability to understand the invention sufficiently to make and use it. For these reasons, and the others below, this claim term is clear and definite, and would not be understood to be a 112(6) term.

80. In the claim language, the claim does not use the word "means". As a result, my understanding is that there is not a presumption that "the graph generator module" is a 112(6) term. I also note that the applicants stated claims in means-plus-function claim language when they intended to invoke § 112(6). For example, unasserted claim 57 shown below includes at least three means-plus-function limitations.[10]

> 57. [pre] A computer-implemented apparatus for generating data graphical displays, comprising:
>
> [57.1] ***means for receiving*** data to be displayed in a non-textual format, said received data being indicative of a plurality of variables;
>
> [57.2] ***means for retrieving*** graph style data items from a data file,

[10] I have been informed that SAS originally asserted, but had subsequently withdrawn, claim 57 having these different means-plus-function limitations.

> [57.3] said graph style data items containing display characteristics to be used in displaying the data in a non-textual format; and
>
> [57.4] said graph style data items containing graph style metadata that specify statistical roles that different data variables have within the data;
>
> [57.5] ***means for accessing*** of the graph style data items by different types of software applications ***in order to display*** through the different types of software applications ***non-textual formatted output based upon the graph style metadata.***

('519 Patent, claim 57, emphasis added).

As noted by the emphasized text above, unasserted claim 57 includes similar functions that Defendants identify when alleging that "the graph generator module" is a 112(6) term. (Specifically, Defendants proffer "receives data to be displayed in a non-textual format" as their construction of one of the functions to be performed by this claim term; unasserted claim 57 contains "means for receiving.") The applicants clearly knew how to draft claims and claim terms (including those having the same or similar functions) in 112(6) format using the word "means" if they desired. However, in claims 34, 52, 53 and 58, the applicants chose not to use the words "means for". Moreover, claim 52 provides a structural limitation, specifying that "the graph generator module is located in a second computer-based storage medium location." This further corroborates that a POSITA would understand and recognize a "graph generator module" to be a well-known software component whose internal details are not germane to a POSITA's ability to make and use the invention.

81. In addition, the only limitation in claim 34 that identifies a function for the graph generator module does not identify it as a 112(6) term. The only such limitation is [34.5]:

> [34.5] said ***graph generator module generating at least one graphical output based upon the received data***, said graphical output being generated in accordance with the defined data characteristics of the graph styles data structure.

I understand the term "generating at least one graphical output based upon the received data" to be functional, but this does not indicate that the graph generator module is a 112(6) term. Instead, it only narrows this limitation by requiring the graph generator module to generate a graph in a particular way. The same is true for claim 58.

82. Because Defendants are incorrect that "the graph generator module" is a 112(6) term, they are incorrect that the term is indefinite due to lack of corresponding structure.

83. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants also state that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding the lack of corresponding structure for this term." This statement does not state that Dr. Jones will opine on why the term "the graph generator module" should be considered a 112(6) term in the first place. Regardless, this statement does not provide any indication of why a POSITA would understand this to be a 112(6) term, or why a POSITA would conclude that there is no corresponding structure for this term even assuming it was a 112(6) term despite the detail disclosed in the specification.

84. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "the graph generator module" reasonably informs the POSITA about the scope of the claim. A POSITA would not conclude that this term is a 112(6) term.

B. The '686 Patent Claims and Disputed Claim Terms

85. Following are representative asserted claims from the '686 Patent. The claim terms that Defendants assert are indefinite are shown in ***bold italics***.

> 1. [pre] A computer-implemented method for handling a database statement from a first database system, comprising the steps of:
>
> [1.1] receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

[1.2] accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

[1.3] generating a second fourth-generation language database statement that is used within a second database system,

[1.4] wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data,

[1.5] wherein the second database statement is compatible with the second database system's query language format;

[1.6] wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement

[1.7] wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

[1.8] using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

[1.9] wherein a first component software object is associated with a first logical piece contained in the tree; ***wherein the first component software object is associated with a first method to textualize***, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

[1.10] using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

[1.11] ***wherein a first software driver textualizes through a second method*** a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

[1.12] ***wherein a second software driver textualizes through a third method*** a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

> [1.13] switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;
>
> [1.14] wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.
>
> * * *
>
> 26. The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.
>
> * * *
>
> 27. The method of claim 26 wherein the first database system's query language format utilizes ***a superset of the SQL standard***.
>
> * * *
>
> 45. The method of claim 1 ***wherein the switching of the association includes switching pointing of the first method to 10 the second method for the first software driver.***

The following is my claim construction analysis for the terms above. As discussed below, none of these terms is indefinite, and a POSITA would not have a firm conviction based on the intrinsic and extrinsic evidence that any of these terms are indefinite.

i. "a superset of the SQL standard"

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"a superset of the SQL standard" (claims 27, 28)	Plain and ordinary meaning	Indefinite

86. The claim term "a superset of the SQL standard" appears in dependent claims 27 and 28. The meaning of this claim term is clear and definite, including from both the plain language and the specification. Claim 27 in its entirety states:

> 27. The method of claim 26 wherein the first database system's query language format utilizes ***a superset of the SQL standard***.

87. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that this term is "[i]ndefinite," but do not provide any explanation for why this term is allegedly indefinite. I understand that Defendants have now proposed a construction of this term as "a set that includes all of the SQL standard and additional elements not in the SQL standard." This change does not cause me to alter my testimony, below. This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how it is not clear how a database system could use a 'superset' of the SQL standard but also have variations in the standard such that it does not use the entirety of the standard language format." From this statement, Defendants apparently misunderstand the use of "superset" and related terminology in the claims and in the specification. Yet, as I will show, these are clear and unambiguous to a POSITA.

88. A POSITA understands that "superset" in the context of programming languages borrows from the term in mathematics, wherein a superset of S is defined as a set that contains all of the elements of S plus possibly additional elements (a set can be a superset of itself). In the programming language context, a superset of a language L contains all of the syntactic and semantic constructs of L plus possibly others. This is consistent with the definition of "superset" given in *McGraw-Hill Dictionary of Scientific and Technical Terms, Sixth Edition* (2003), which is: "A programming language that contains all the features of a given language and has been expanded or enhanced to include other features as well." A copy of this dictionary definition is attached hereto as **Exhibit B**. Claims 27 and 28 refer to "superset[s] of the SQL standard,"

meaning database query languages that have at least the constructs of standard SQL[11] plus possibly others.

89. The specification, meanwhile, presents disclosures that are broader than the claims—disclosures that involve *dialects* of SQL in addition to *supersets* of SQL. In the programming language context, a POSITA would understand "dialect" of a language to mean a language is one that has many but not necessarily all of the constructs of that language and possibly also others, just as one human language is a dialect of another (e.g., Quebecois is a dialect of French or Plattdeutsch is a dialect of German). In other words, "dialect" is a more general case than "superset": supersets of a language L are dialects of L but not necessarily vice versa. This is consistent with the definition of "dialect" in *Microsoft Press Computer Dictionary, Third Edition* (1997), which is: "A variant of a language or protocol. For example, Transact-SQL[12] is a dialect of structure query language (SQL)." A copy of this dictionary definition is attached hereto as **Exhibit B**.

90. The specification uses the terms "superset" and "dialect" consistently with these meanings. It says:,

> "Within the system 30, a textualization process 50 addresses the complexity of translating a native database statement 32 ***dialect*** into a variety of third party database dialects (34, 36, 38) by allowing the common parts of the default syntax of functionality to be shared between a native database and a third party database. The textualization process 50 utilizes database specific textualizations 52 to translate the common parts to the third party database ***dialect***."

[11] At the time when the application that would issue as the '686 Patent was filed, the SQL standard in force was SQL:1999, published in December 1999. *See for example* https://www.iso.org/standard/26196.html and https://www.iso.org/standard/26197.html. The standard is colloquially referred to, including in the '686 Patent, as "ANSI SQL," because the (American) ANSI standards body created the original SQL standard in 1986 before subsequently handing the standard off to (the international) ISO, which manages it to this day.

[12] Transact-SQL is the dialect of SQL supported by Microsoft and Sybase database systems.

> ('686 Patent at 2:24-26, emphasis added);

and:

> The textualization process 50 compartmentalizes an SQL statement 32 into logical text pieces or components which are initially provided based on a default SQL ***dialect*** … Any of these text pieces can be overridden by a third party SQL provider that utilizes a different SQL ***dialect*** than the default.
>
> (*Id.* at 3:12-17, emphasis added)

Regarding supersets, it says:

> "It should be further understood that different types of databases refer to database systems that contain differences in their respective database statement format and/or syntax, ***such as utilizing a different superset*** of an ANSI database statement standard."
>
> (*Id.* at 2:49-53, emphasis added.)

91. In other words, the specification is stating that different database can use different query languages, *some of which* may be different supersets of the standard SQL language.[13] And the examples given in the specification involve *dialects* of SQL that are not necessarily *supersets* of the standard; see for example *Id*. at 4:1-11, 4:43-50, 7:62-67, and 8:26-64. Only claims 25 and 27 limit their scope to *supersets* of standard SQL. Thus, there is no ambiguity about the use of "superset" in the claims and that the specification embodies that scope. It is my understanding that the specification may disclose functionality beyond that which is recited in the claims.

[13] The Summary section of the specification states that "… most database systems, such as those from Oracle, Sybase, Business Object, SAS, or Brio, implement a superset of the ANSI standard." ('686 Patent at 1:16-19.) Those commercial systems were known to maintain compatibility with official SQL standards over time, in addition to implementing their own dialects or supersets of the standards. Regardless of whether the foregoing statement was true at the time of filing, it is also consistent with the specification's use of the term "superset" and does not limit the specification's applicability to supersets of standard SQL.

92. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "a superset of the SQL standard" reasonably informs the POSITA about the scope of the claim and is not indefinite.

ii. [withdrawn]

[Paragraphs 93-96 withdrawn]

iii. "wherein the first component software object is associated with a first method to textualize"

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"wherein the first component software object is associated with a first method to textualize" (claim 1)	Plain and ordinary meaning	Indefinite

97. This claim term appears in independent claim 1. The meaning of this claim term is clear and definite, including from the claim language itself. Once again, the claim means what it says. Earlier limitation [1.6] of claim 1 refers to a "first component software object" that is "associated with a first logical piece contained in a tree." ('686 Patent, 10:24–25). This generally refers to the fact that each logical piece of a tree representation of the first database's syntax has a software object associated with it. Subsequent limitation [1.9] at issue then states that "… the first component software object is associated with a first method to textualize," meaning that this software object is associated with a first method to textualize from a certain aspect of the native database syntax. This is apparent from the complete context of this claim term within limitation [1.9], which is as follows:

> [1.9] wherein a first component software object is associated with a first logical piece contained in the tree; ***wherein the first component software object is associated with a first method to textualize***, into fourth-generation database language text, the first

> component software object's associated logical piece that is
> contained in the tree;

When read in its entirety, this limitation states that the software object is associated with (1) a first logical piece contained in the tree (i.e., a piece of the native syntax statement from the first database system) and (2) a first method to textualize the software object's "associated logical piece that is contained in the tree" into fourth-generation database language text.

98. This definite meaning to a POSITA is confirmed by the specification, which explains that "[t]he textualization process 50 compartmentalizes [each] SQL statement 32 into logical text pieces or components which are … represented in the SQL tree." (*Id.*, 3:12–15). "[A]n SQL tree is used by the textualization process 50 to process an SQL statement 32. The SQL tree 60 represents the syntax of a native database's SQL statement 32 and its related metadata … ." (*Id.*, 3:1–5). "[A] driver object 110 is responsible for creating an 'override' 112 to the default method 106" for textualizing an SQL component object statement. (*Id.*, 3:61–67). In other words, a component software object includes a textualization method that reflects the syntax of that component of the native database's SQL syntax, as is the case here for the claimed "first component software object" claim term in dispute.

99. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that this term is "[i]ndefinite," but do not provide any explanation for why this term is allegedly indefinite. This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how it is not clear what is meant by a first method to textualize, or how the switching of the association between the first component software object and the first method to textualize includes switching of the first method to the second method for the first software driver." This statement is unclear about Defendants' contention why this term is indefinite. Regardless, this term is not indefinite as explained above.

100. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "wherein the first component software object is associated with a first method to textualize" reasonably informs the POSITA about the scope of the claim and is not indefinite.

iv. "wherein a first software driver textualizes through a second method"

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"wherein a first software driver textualizes through a second method" (claim 1)	Plain and ordinary meaning	Indefinite

101. This claim term also appears in independent claim 1. The meaning of this claim term is clear and definite, including from the claim language itself. Once again, the claim means what it says. This claim term appears in limitation [1.11], which states:

> [1.11] ***wherein a first software driver textualizes through a second method*** a logical piece into fourth-generation database language text that is compatible with the second database system's query language format

As shown by this claim language, this limitation requires "a first software driver" that "textualizes through a second method" a logical piece of the tree representation of the first database language syntax and associated metadata. (*See* '686 Patent, limitation [1.4]). Further, this limitation requires that the textualization to be "into a fourth-generation database language text that is compatible with the second database system's query language format." (*Id.*, limitation [1.11]). Thus, the claim means what it says, requiring "a second method" that will textualize a "logical piece" into the syntax of the second (non-native) database system's query language format.

102. This definite meaning to a POSITA is confirmed by the specification. As already discussed, the specification explains that "[t]he textualization process 50 compartmentalizes

[each] SQL statement 32 into logical text pieces or components which are … represented in the SQL tree." (*Id.*, 3:12–15). "[A]n SQL tree is used by the textualization process 50 to process an SQL statement 32. The SQL tree 60 represents the syntax of a native database's SQL statement 32 and its related metadata … ." (*Id.*, 3:1–5.) "[A] driver object 110 is responsible for creating an 'override' 112 to the default method 106" for textualizing an SQL component object statement. (*Id.*, 3:61–67.) In other words, a component software object can also include a textualization method that reflects the syntax of that component of the non-native database's SQL dialect, as is the case here for the claimed "second method" of the "first software driver" claim term in dispute. By textualizing using the "second method" (i.e., the method related to the syntax of the non-native database) rather than the "first method" (i.e., the method related to the syntax of the native database), SQL statements are textualized and transformed into the required syntax for the non-native database.

103. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that this term is "[i]ndefinite," but do not provide any explanation for why this term is allegedly indefinite. This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how it is not clear how or if the first software driver textualizes through a second method at least because the claim also states that because of the switching of the association of the first component software object, the first component software object textualizes through the second method." It is not entirely clear what Defendants mean or what their indefiniteness theory is through this brief statement. Regardless, there is no inconsistency or indefiniteness from this claim term and its meaning described above and the other claim language in claim 1. For the reasons explained above, this claim term is not indefinite.

104. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "wherein a first software driver textualizes through a second method" reasonably informs the POSITA about the scope of the claim and is not indefinite.

v.<u>"wherein a second software driver textualizes through a third method"</u>

Term (Claim)	Plaintiff's Proposed Construction	Defendants' Proposed Construction
"wherein a second software driver textualizes through a third method" (claim 1)	Plain and ordinary meaning	Indefinite

105. This claim term also appears in independent claim 1. The meaning of this claim term is clear and definite, including from the claim language itself. Once again, the claim means what it says. This claim term appears in limitation [1.12], which states:

> [1.12] ***wherein a second software driver textualizes through a third method*** a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

As shown by this claim language, this limitation requires "a second software driver" that "textualizes through a third method" a logical piece of the tree representation of the first database language syntax and associated metadata. (*See* '686 Patent, limitation [1.4]). Further, this limitation requires that the textualization to be "into a fourth-generation database language text that is compatible with the third database system's query language format." (*Id.*, limitation [1.12]). Thus, the claim means what it says, requiring "a third method" that will textualize a "logical piece" into the syntax of the third (non-native) database system's query language format. The additional requirement of this limitation is the introduction of a second non-native database, and a corresponding "second software driver" having a "third method" for textualizing corresponding to this second, additional non-native database. In other words, the invention as a whole covers textualizing and transforming a native SQL statement in one format into at least two non-native SQL statement formats for two different non-native databases.

106. This definite meaning to a POSITA is confirmed by the specification. As already discussed, the specification explains that "[t]he textualization process 50 compartmentalizes [each] SQL statement 32 into logical text pieces or components which are … represented in the SQL tree." (*Id.*, 3:12–15.) "[A]n SQL tree is used by the textualization process 50 to process an SQL statement 32. The SQL tree 60 represents the syntax of a native database's SQL statement 32 and its related metadata … ." (*Id.*, 3:1–5.) "[A] driver object 110 is responsible for creating an 'override' 112 to the default method 106" for textualizing an SQL component object statement. (*Id.*, 3:48–67.) In other words, a component software object can also include a textualization method that reflects the syntax of the non-native database's SQL statements, as is the case here for the claimed "third method" of the "second software driver" claim term in dispute. By textualizing using the "third method" (i.e., the method related to the syntax of an additional, second non-native database) rather than the "first method" (i.e., the method related to the syntax of the native database), SQL statements are textualized and transformed into the required syntax for a second, additional non-native database.

107. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that this term is "[i]ndefinite," but do not provide any explanation for why this term is allegedly indefinite. This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how it is not clear how or if the second software driver textualizes through a third method." It is not entirely clear what Defendants mean or what their indefiniteness theory is through this brief statement. As discussed above, the claim language

clearly states that the second software driver textualizes through a third method. Regardless, for the reasons explained above, this claim term is not indefinite.

108. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "wherein a second software driver textualizes through a third method" reasonably informs the POSITA about the scope of the claim and is not indefinite.

C. The '996 Patent Claims and Disputed Claim Terms

109. Following is a representative asserted claim from the '996 Patent. The claim terms that Defendants assert are indefinite are shown in ***bold italics***.

> 19. [pre] A computer implemented system for processing a query, comprising:
>
> [19.1] one or more physical data processors;
>
> [19.2] one or more non-transitory computer-readable storage mediums containing instructions configured to cause the one or more processors to perform operations including:
>
> [19.3] receiving a native syntax query requesting data stored in a non-native database that uses a ***non-native*** syntax, wherein the query is received at an application that is separate from the non-***native*** database, wherein the query requests that the data be retrieved from the ***non-native*** database, wherein the query requests that a processing operation be performed on the requested data by the application, wherein the query includes one or more expressions, and wherein one or more of the expressions includes one or more functions;
>
> [19.4] parsing the ***native*** syntax query, wherein parsing includes identifying a function within an expression that cannot be processed by the ***non-native*** database, wherein the function specifies the processing operation to be performed on the requested data by the application, wherein a plurality of labels are associated with the function and the expression, and wherein labels include constant labels and format labels;
>
> [19.5] analyzing the function and the expression to determine a context of the function within the expression, wherein the context describes how the function is used within the expression;

> [19.6] generating a final expression query by obtaining a control string from an internal table for each of the plurality of labels associated with the function and the expression, wherein label modifiers are applied to format labels;
>
> [19.7] transforming the ***native*** syntax query into an equivalent ***non-native*** syntax query, wherein transforming includes parsing and inserting the final expression query into the equivalent ***non-native*** syntax query using the function, the expression, and the context to translate the function and the expression into multiple functions and multiple expressions that are configured for processing by a ***non-native*** database system;
>
> [19.8] transmitting the equivalent ***non-native*** syntax query to a ***non-native*** database system to generate results and to perform the processing operation on the generated results;
>
> [19.9] receiving processed results from a ***non-native*** database system; and
>
> [19.10] transmitting the processed results to a client application

The following is my claim construction analysis for the terms above. As discussed below, none of these terms is indefinite, and a POSITA would not have a firm conviction based on the intrinsic and extrinsic evidence that any of these terms are indefinite.

i. “native” / “non-native”

Term (Claim)	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“native” (claims 1, 2, 3, 19, 20, 23, 37, 38, 41) “non-native” (claims 1, 5, 19, 23, 37, 41)	Plain and ordinary meaning	Indefinite

110. The claim terms “native” and “non-native” appear throughout the asserted claims as modifiers to nouns such as database and database system, and syntax and syntax query. The

meanings of the terms "native" and "non-native" in the claim terms are clear and definite, including from the claim language itself.

111. "Native" simply means "the type of system a user or program is working with," as opposed to other types of systems. This is the plain and ordinary meaning of the term as understood by a POSITA. The analogy, of course, is to "native language" as opposed to "non-native language." This comports with contemporaneous dictionary definitions of "native," such as:

a. *The Facts On File Dictionary of Computer Science* (2001) defines "native compiler" as "[t]he usual form of compiler that runs on a computer producing an object file for that computer, as opposed to a cross compiler."[14]

b. *Webster's New World Computer Dictionary, Tenth Edition* (2003) defines "native compiler" as "a compiler that is designed to execute on the computer for which it is generating code."

c. *Barron's Dictionary of Computer and Internet Terms, Tenth Edition* (2009) defines "native" as "Designed for a specific hardware or software environment (rather than for compatibility with something else). "

d. *Microsoft Press Computer Dictionary, Third Edition* (1997) defines "native application" as "A program that is designed specifically for a particular type of microprocessor, that is, a program that is binary compatible with a processor." It defines "native code" as "Code specific to a particular machine or processor", and it defines "native compiler" as "A compiler that produces

[14] A compiler is a program that translates code written (by a programmer) in a programming language into machine code for a computer to run, such as a C++ compiler for Windows PCs. A cross-compiler is a compiler that produces machine code to be run on a different type of computer from the one running the compiler, such as a C++ compiler for Windows PCs that produces code for an Apple Macintosh.

> machine code for the computer on which it is running, as opposed to a cross-compiler, which produces code for another type of computer."

Copies of these dictionary definitions are attached hereto as **Exhibit B.**

112. In the context of the '996 Patent, a POSITA would understand that "native" refers more specifically—while not diverging from the plain and ordinary meaning—to the query language syntax supported by the database that a client application is accessing.[15] That is, a user working with a given database system would use its own ("native") querying capability to submit queries, and the same query language syntax could be used to submit queries to another database that supports the same language. A database system that uses a query syntax other than the one with which the user is working is thus a "non-native" system. A "native" SQL statement may not necessarily be supported by a "non-native" database system that that supports SQL statements written in a different, "non-native" syntax. As a result, if an application program seeks data stored in a "non-native" database system, any SQL statements that generated by that application program in a "native" syntax must be transformed into a "non-native" syntax that can be used to query the "non-native" database system.

113. This understanding of a POSITA is consistent with the specification. The summary of the invention explains that the invention is "for generating a query in a first fourth generation language at a native system." ('996 Patent, 1:59–60). However, if "[t]he query is intended for a non-native database system capable of processing queries in a second fourth generation language," then "the query is transformed through the use of formats into an equivalent query expression that can be processed by the non-native database system." (*Id.*,

[15] Other uses for this term that a POSITA would understand include "native code" generated by a programming language compiler for the same type of system that the compiler runs on.

1:60–2:4; *see also id.*, 2:4–8.) As stated in this excerpt and elsewhere, the specification explains that the "native" database will support one fourth generation (4GL) programming language, such as the SAS language, whereas the "non-native" database and syntax will support a second, different fourth generation programming language. (*Id.*, 4:31–59.) The specification also provides various examples using the SAS language "put( ) function," which is supported if querying a "native" SAS database but may not be supported if querying a "non-native" third-party database "such as Oracle, DB2, Teradata, and others." (*See id.*, 3:41–4:25.) The specification discloses a "transformation system" for transforming a "native" statement such as one that includes SAS's "put( ) function" into a "non-native" statement that can be used to retrieve and process data from a "non-native," non-SAS third-party database system. (*See generally id.*, 4:1–10:49 and Figs. 4–8 (describing system for transforming "native" SQL statements into "non-native" SQL statements for accessing a non-native database system); 10:50–16:42 (disclosing ten different examples of transformation depending upon the context of the "put( ) function" within a larger SQL expression)).

114. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants assert that the terms "native" and "non-native" are "[i]ndefinite." This disclosure also states that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding how a person of skill in the art would not understand this term, including because it is unclear what the terms native and non-native mean in this context, or what each is native or non-native in relation to." Contrary to this statement, as explained above, it is clear what "native" and "non-native" mean and relate to in the context of the '996 Patent.

115. For the reasons discussed above, it is my opinion that a POSITA would find that the claim terms "native" and "non-native" reasonably inform the POSITA about the scope of the claim and is not indefinite.

D. The '458 Patent Claims and Disputed Claim Terms

116. The following are representative asserted claims from the '458 Patent. The claim term that Defendants assert is governed by § 112(6) is shown in ***bold italics***.

> 1. [pre] A model repository system, comprising:
>
> [1.1] a data store for storing a plurality of data records;
>
> [1.2] a data mining application for analyzing the data records and for generating a plurality of data models; and
>
> [1.3] a model repository for storing the generated data models, wherein the model repository includes one or more index structures containing a plurality of attributes associated with the data models;
>
> [1.4] wherein the data models are predictive data models;
>
> [1.5] wherein the predictive data models are the entities being indexed by the one or more index structures such that the attributes of the predictive data models are stored within the one or more indexes;
>
> [1.6] a ***model repository facility*** for exporting the generated data models to the model repository;
>
> [1.7] at least three configuration files stored in the model repository, wherein a first configuration file stores information that is used by the model repository facility in exporting the generated data models to the model repository, and second and third configuration files store information that is used by the model repository system in building the main index in the model repository from attributes supplied by human end users and from the data mining application.
>
> ***
>
> 27. [pre] The model repository system of claim 1, wherein:

> [27.1] the ***model repository facility*** builds the index structures stored in the model repository after one or more selected models have been exported to the model repository.
>
> ***
>
> 61. [pre] A model repository system, comprising:
>
> [61.1] a data store for storing a plurality of data records;
>
> [61.2] a data mining application for analyzing the data records and for generating a plurality of data models; and
>
> [61.3] a model repository for storing the generated data models, wherein the model repository includes one or more index structures containing a plurality of attributes associated with the data models;
>
> [61.4] a ***model repository facility*** for exporting the generated data models to the model repository;
>
> [61.5] a first configuration data store that stores information to be used by the model repository facility in exporting the generated data models to the model repository; and
>
> [61.6] a second configuration data store that stores information to be used by the model repository system in building the one or more indexes in the model repository.

The following is my claim construction analysis for the terms above.

i.“model repository facility”

Term (Claim)	Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“model repository facility” (claims 1, 3, 9, 27, 61, 64, 70, 88)	Not governed by 35 U.S.C. § 112(6)	35 U.S.C. § 112(6) Claims 1, 3, 61, 64: Function: exporting the generated data models to the model repository Structure: Figs. 7A through 7C. Claims 27, 88:

		Function: building the index structures stored in the model repository. Structure: Figs. 8A through 8C.

117. This term appears in independent claims 1 and 61, and dependent claims 3, 9, 27, 64, 70, and 88. In Defendants' '458 4-2 Disclosures, Defendants propose to treat this term under 35 U.S.C. § 112(6), i.e., as a means-plus-function term. Defendants further propose to treat this single claim term as having two different functions recited in two different sets of claims. Defendants specifically proffer the function "exporting the generated data models to the model repository" for the term as used in claims 1, 3, 61, and 64; and the function "building the index structures stored in the model repository" for the term as used in claims 27 and 88. Defendants cite Figures 7A through 7C as the purported structure supporting the first of of these functions and Figures 8A through 8C as the purported structure supporting the second function. Defendants cite neither function, and indeed offer no construction, for the term's use in claims 9 and 70.

118. In my opinion, "model repository facility" is not a 112(6) term. As an initial matter, the structure of both independent claims 1 and 61 include a number of elements with parallel grammatical structures ("a data store for storing …", "a data mining application for analyzing …", "a model repository for storing …" and "a model repository facility for exporting …"). None of these formulations use "means" and therefore there is not a presumption that they are 112(6) terms. Indeed, Defendants have accepted that none of the other limitations in these claims are 112(6) terms, and it would be inconsistent to conclude that "model repository facility" alone is subject to 112(6).

119. Further, it is clear from the specification that "facility" is used to refer to a software program or portion thereof. *See, e.g.,* Abstract ("…a model repository facility that is preferably integrated into the data mining application …"); col. 2:46 ("Associated with the model repository is a model repository facility that is preferably integrated into the data mining application…"); col. 3:62 ("The data mining application 18 preferably includes an integrated model repository facility (MRF) application 18A…); col. 3:67 ("Alternatively, however, the MRF 18A could be a stand-alone application, in which case it would not be integrated into the data mining application 18."). This understanding is also supported by claims 9, 10, 70 and 71, which recite, alternatively, that the model repository facility is "integrated into the data mining application" (claims 9 and 70) or "a stand-alone software application" (claims 10 and 71). In addition, during prosecution, the Examiner clearly understood the "model repository facility" to be a software component. *See, e.g.,* 12/27/02 Office Action at pp. 5, 7; 7/31/03 Office Action at pp. 4, 5; 2/23/04 Office Action at pp. 4-5; 11/23/04 Office Action at pp. 7, 9 (all referencing the model repository facility as a stand-alone software application or integrated into the data mining application). Thus, it would be clear to a POSITA that "model repository facility" is not a nonce phrase, but in fact references a software application, or computer program, that is either stand-alone or incorporated into another application; based on my understanding of the legal standards as set forth above in ¶¶22-28, such a software application would not, as claimed, be subject to 112(6) or be understood by a POSITA as a means-plus-function claim. Moreover, the purported functions, "storing" and "exporting," are well within the range of standard software functional descriptions that, I understand, do not trigger application of 112(6).

120. In their amended Exhibit A to their P. R. 4-2 disclosure, Defendants also state that "Defendants reserve the right to rely on the expert testimony of Dr. Mark Jones regarding the

meaning of this term, its function, and the corresponding structure, if any, described in the specification." This statement does not state that Dr. Jones will opine on why the term "model repository facility" should be considered a 112(6) term in the first place. Regardless, this statement does not provide any indication of why a POSITA would understand this to be a 112(6) term.

121. For the reasons discussed above, it is my opinion that a POSITA would find that the claim term "model repository facility" reasonably informs the POSITA about the scope of the claim. A POSITA would not conclude that this term is a 112(6) term.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Dated: November 17, 2019

Bill Rosenblatt